ProQuest® Return to NPL Web Page ?Help

Collections | Search Methods ▾ | Topic Finder | Browse Lists ▾ | Results ▾ | Search Guide

Searching collection: All Collections Article Display

Email Article Article 11 of 13 Publisher Info.

Print Article □ Mark article Article format: Full Text ▾

Save Link Saves this document as a Durable Link under "Results-Marked List"

Novell feathers its NEST protocol kit

Computerworld; Framingham; Mar 11, 1996; DiDio, Laura;

Duns: **03-778-7298**	
Volume:	30
Issue:	11
Start Page:	57
ISSN:	00104841
Subject Terms:	TCP-IP Remote control Product introduction **Internet** Computer upgrading Computer programming TCP-IP Software Data transmission Computer programming
Classification Codes:	**9190**: *US* **9120**: *Product specific treatment* **5240**: *Software & systems*
Geographic Names:	US
Companies:	Novell Inc*Ticker:***NOVL***Duns:***03-778-7298** Novell Inc

Abstract:

Last week, Novell Inc. released an enhanced version of the Novell Embedded Systems Technology (NEST) Software Developer's Kit that includes native TCP/IP support. NetWare users will be able to connect to the ***Internet*** *by using this standard protocol instead of Novell's proprietary IPX/SPX protocol. NEST is Novell's method of taking core components of the NetWare network operating system and embedding them in consumer products such as fax servers, telephones and* ***home appliances****. The goal is to let home and office users control these via public and private networks.*

Full Text:

Imagine using electric power lines to transfer corporate data. Or automating routine tasks such as setting the of fice temperature or programming the cappuccino machine.

Those capabilities sound far out, but they may become available in the next five years via Novell, Inc.'s Embedded Systems Technology (NES. The company last week released an enhanced version of the NEST Software Developer's Kit that includes native TCP/IP support.

NetWare users will be able to connect to the Internet by using this standard protocol instead of Novell's proprietary IPX/SPX protocol.

Novell also is packaging its NEST Autoroute global document routing in the developer's kit for free. Novell formerly sold the Autoroute facility as a separate package priced at $50,000.

NEST Autoroute allows manufacturers to add global document routing capabilities to fax, copy, scan and print devices. "So theoretically, scanned print documents could automatically be routed to a user's E-mail box," observed Bob Sakakeeny, an analyst at Aberdeen Group, Inc. in Boston.

NEST, which was announced a year ago, is Novell's method of taking core components of the NetWare network operating system and embedding them in consumer products such as fax servers, telephones and home appliances. The goal is to let home and office users control these devices via public and private networks.

Utilicorp United, Inc., in Kansas City, Mo., is an early NEST user and Novell partner. The utility supplies power in 17 Midwest states. Utilicorp is using the software to develop the NEST Powerline technology to let businesses create a smart energy network.

"The smart energy network based on NEST could perform tasks [such as] reading home and office electrical meters. This could eliminate the need for meter readers," Sakakeeny said.

He said embedding a piece of NetWare code in consumer devices to automate "every conceivable task" is still a dream but will likely become reality within five years.

"The market for embedded systems that NEST is targeting is still tiny, but it's good for Novell to do. It lets them establish a beachhead in an emerging industry, and it costs the company next to nothing since NEST relies on the core NetWare [network operating system] kernel," he said.

The NEST Software Developer's Kit 1.2 will ship at the end of March. It costs $50,000.

ProQuest® Return to NPL Web Page ?Help

Collections | Search Methods ▾ | Topic Finder | Browse Lists ▾ | Results ▾ | Search Guide

Searching collection: All Collections Article Display

Email Article ◁ Article 12 of 13 ▷ Publisher Info.

Print Article □ Mark article Article format: Cite/Abstract ▾

Save Link Saves this document as a Durable Link under "Results-Marked List"

Turkey for coach potatoes

USA Today; Washington, D.C; Oct 27, 1995; Maney, Kevin;

UMI Publication No.:	03744827
Start Page:	B2
Document Type:	Commentary
Source Type:	NEWSPAPER
ISSN:	07347456
Subject Terms:	Software Personal computers **Internet** Cooking Appliances
Companies:	Novell Inc
UMI Journal Code:	USA

Abstract:

*Kevin Maney discusses Butterball Turkey's help line on the **Internet** and Novell's unveiling of networking software for PCs called Nest that will allow users to link any **home appliance** to a PC through regular electrical wiring, saying the possibility exists for cooking one's turkey online.*

ProQuest Return to NPL Web Page ?Help

Collections | Search Methods ▾ | Topic Finder | Browse Lists ▾ | Results ▾ | Search Guide

Searching collection: All Collections Article Display

Email Article ◁ Article 64 of 100 ▷ Publisher Info.

Print Article □ Mark article Article format: Text+Graphics ▾

Save Link Saves this document as a Durable Link under "Results-Marked List"

Building a smart online video application

Dr. Dobb's Journal; San Mateo; Dec 1997; Robin Rowe;

Volume: 22
Issue: 12
Start Page: 78-84+
ISSN: 1044789X
Subject Terms: Object oriented programming
Internet
Application programming interface
Video

Abstract:
Rowe describes "Smart ***VCR****" (SVCR), a new type of application that integrates hard-to-master* ***Internet*** *programming paradigms to create a video-recording technology that allows real-time searches of broadcast TV.*

Full Text:

[Headnote]
A software VCR using off-the-shelf parts

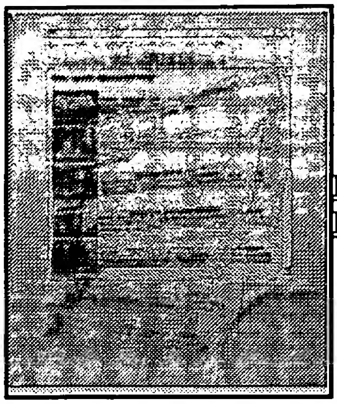
Many of today's applications involve Internet protocols, MPEG-1 video, JPEG images, ActiveX components, Netscape plug-ins, and other hardto-master programming paradigms. Each of these technologies is complex, making it a challenge to build real-time systems that integrate all (or even some) of them.

In this article, I'll describe a new type of application I call "Smart VCR" (SVCR), which integrates these disparate technologies. SVCR is a video-recording technology that enables real-time searches of broadcast television, thereby combining the ubiquity of existing TV channels with the convenient interactive interface of a web browser. An SVCR watches TV for you, and can send a notification to you by e-mail or pop up a window when it finds a topic that is interesting to you.

SVCR captures the closed-captioned text that is encoded in most U.S. television broadcasts and converts that text into program transcripts formatted in HTML. As it scans the closed-captioned text, SVCR searches for clues as to when video segments start and end. In the process, it stores the closed-captioned text into a realtime database and snaps JPEG thumbnail images of the streaming video to become icons and filmstrips. Finally, it captures MPEG-1 video clips at a nominal data rate of about ten MB per minute. The result is a web page (like Figure 1) that lists TV news stories in chronological order, with links to the full transcripts, filmstrips of the video images, and hyperlinks to the actual video clips. A Netscape plug-in or CGI program provides a browser interface to the transcript database, enabling users to keyword search the transcripts, retrieving associated images and video.

SVCR Overview

While an SVCR could be implemented in; many ways, my design has four major hardware components:



Enlarge 200%
Enlarge 400%

* A Pentium-based PC running Windows 95.

* An MPEG-1 encoder card.

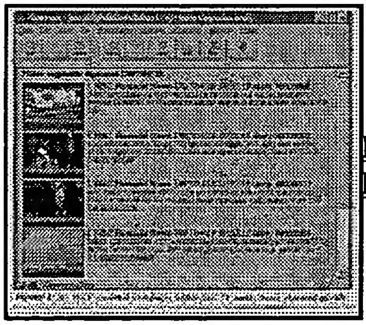
* A JPEG encoder device.

* A closed-captioned decoder box.

Although its name suggests otherwise, an SVCR isn't necessarily a VCR. Over the past several years, the video post production industry has been moving away from video tape to nonlinear editing systems such as the Avid (http://www.avid.com/) and Media 100 (http://www.medial00.com). One advantage of nonlinear systems is their ability to randomly access video clips. It's like the difference between finding a song on a cassette tape and finding one on a CD-you never need to wait for a nonlinear system to rewind or fast forward. In building an SVCR, my approach was to go digital, recording video straight to disk using an MPEG encoder board.

In the interest of real-time performance, all encoding and decoding operations are hardware assisted. Using commercial offthe-shelf hardware, I wrote C++ code to interface with vendor-supplied Windows COM and OCX objects that manipulate the hardware encoders. A separate Windows thread listens to the closed-captioned text as it is delivered on the RS232 port.

SVCR runs in real time either standalone in Windows 95 or on a distributed networked heterogeneous environment. It is both multiprocess and multithreaded. Because Windows 95 supports a variety of commercial graphics hardware, the code to handle digitizing video content was written for that operating system, using MFC with components built in both Symantec C++ and Microsoft Visual C++. To distribute the load of actually serving the video, parts of SVCR were ported to DEC Alpha and Sun Sparc systems, and compiled with GNU g++. The supported client platforms for viewing captured video include Windows 95/NT, Macintosh, and the UNIX systems of DEC, Sun, HP, and SGI.



Enlarge 200%
Enlarge 400%

Figure 1:

Additionally, a database system was written to optimize for minimum latency on writes. The goal was to maintain realtime capture performance on a Pentium 133. For improved performance on user queries, the data may be optionally saved across an Ethernet network using SAMBA to a UNIX box (such as a DEC AlphaStation Server running OSF) rather than to the local disk. (SAMBA is a suite of programs that lets clients access a server's filespace and printers via SMB protocol; for more information, see http://samba .canberra.edu.au/pub/samba/.)

Enter the Culling Agent

Because digital video can consume massive amounts of disk space in a short amount of time (MPEG-1 consumes about two GB every three hours), a "culling agent periodically removes video that isn't of interest; see Figure 2. Without a culling agent, you must create a large central repository for housing row upon row of RAID drives. Even if you are willing to store it that way, delivering full-quality MPEG-1 video requires more network bandwidth than is available on the Internet today. Collecting the video content through the existing broadcast channels and saving it locally sidesteps this bandwidth bottleneck. (The video repository approach is being researched through the DARPA-sponsored Digital Library Initiative; see http://www .aero.hq.nasa.gov/hpcc/cdrom/content/reports/annrpt96/ta/JRI.htm.)

Why not limit the recording of video segments at the time of capture, instead of culling later? Because this would return us to the same problem a human has when deciding whether or not to record a program. It doesn't usually become apparent that it is an interesting segment until it is already half over. Even if such aggressive culling was possible, it wouldn't be desirable because it leaves no window for users to change their minds about what's worth saving.

A Serial Port Thread in MFC Each frame of video that is closed captioned contains two text characters; even if they are just nulls (padding). The Telescriber closed-captioned decoder box is an off-the-shelf device primarily intended for the hearing impaired (see http://www.viewcomtech.com/). The Telescriber reads the closed captioning hidden in the retrace interval of line 21 of the video signal and outputs it into a standard serial port. The first step in developing SVCR was to integrate this device by writing my own program to listen to it.

The closed-captioned text transmitted in a TV signal isn't ASCII, but its own code that is set down by FCC rule 91-119, mod 92-157. (It can be found in the Code of Federal Regulations Title 47, Part 15, Section 119.) Stripping off the top bit of closed-captioned text converts it to a rough approximation of ASCII. It's only an approximation because the closedcaptioned text contains formatting codes in addition to data. It also has a somewhat different symbol set; for example, a musical note to represent singing.

To monitor the serial port in Windows, you start a separate thread of execution devoted to this task. (Code to control the serial port in Windows is widely available. See, for instance, Programming Windows 95 Unleashed, edited by Randall A. Tamura, Sams, 1995.) The fundamental approach is to use a CWinThread in conjunction with OVERLAPPED event objects.

In Windows 95, the serial port is opened using CreateFile() and treated as an asynchronous file. Windows NT supports asynchronous files (which read or write in the background), but Windows 95 does not. Windows 95 only supports them if the "file" is actually a port. The serial port thread fills a buffer with data while the main thread empties it. The serial thread notifies the main thread that data is waiting by using view->PostMessage().

Using COM, OLE, and OCX

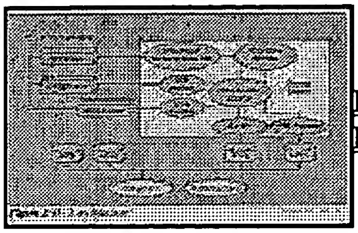
SVCR talks to two other hardware devicesa Snappy JPEG image-capture device (http://www.play.com/), which hangs off the parallel port, and an MPEGator MPEG-1 encoder card (http://www.darim .corn/) installed in a PCI slot. The Snappy includes an API controlled through an OCX. The MPEGator is controlled through a COM interface. Although an OCX is a type of COM, programming to these two interfaces is very different.

COM is the underlying mechanism that allows Windows applications to interface with objects that are external to a program. To a C++ programmer, this means that you get a pointer to an object in an external program, with all the power that implies. ActiveX and OCX controls are standard interfaces that use COM. Similar in concept to abstract base classes, these interfaces are the methods that a particular type of COM class must provide.

ActiveX was formerly known as OLE, and you will still find references to OLE in documentation. The OCX interface evolved from the VBX control. It's a COM interface intended for use by Visual Basic, but available to other languages, too, since it is just another kind of COM. ActiveX is a more lightweight interface than OCX (and is, consequently, more popular for writing ActiveX controls that will be downloaded via the Internet).

Look Ma, No Libs!

Not long ago, to communicate with a vendor's proprietary hardware, you made some calls into its programming API and linked your code with their C library, which implemented low-level control of their device. More recently, an enhancement came in the form of Windows Dynamic Link Libraries (DLLs). You don't have to link statically anymore because you can load the (vendor's) DLL at run time. The Windows COM interface, of which ActiveX and OCX are examples, goes a step further. The COM mechanism lets you retrieve the vendor's library as though it was a C++ object. The COM object is actually an executable or a DLL, but that detail is hidden.



Enlarge 200%
Enlarge 400%

Figure 2:

You can retrieve a COM object either by its CLSID number (its unique Windows ID number generated at compile time) or by name (using the Windows registry).The compiler's class wizard generates MFC code to make this operation invisible to the programmer.

Since ActiveX is smaller than OCX, it would be preferable in a real-time program to use ActiveX. Snappy offers a COM interface, but that only handles raw bitmaps. Capturing a JPEG image is part of their OCX interface. It was a further disappointment that Symantec C++ 7.2 (the Windows C++ compiler I first used) doesn't support controlling an OCX. This was confusing because Symantec C++ does support creating an OCX.

As a workaround, I created the code to control the Snappy OCX using VC++ and wrappered it inside a new ActiveX control. I controlled that from my Symantec C++-compiled application. After Symantec C++ 7.5 came out, which included support for the latest version of MFC, I was able to migrate the VC++ code back into the main application. I now build everything with both VC++ and Symantec C++ so I can use either vendor's tools.

Working with an OCX or ActiveX control is almost trivial with C++ compiler support. Using the class wizard, the compiler generates a class with the code to handle the COM communication. Using an object of this class, you can drive the external control as you would if it was an object compiled into your own code.

The MPEGator is a popular MPEG-1 video-capture card. A high-quality MPEG-1 datastream consumes ten MB per minute, a low-resolution stream about two MB per minute. The MPEGator also supports AVI, but that consumes more space for the same quality. Listing One (listings begin on page 103) is a simple test program that saves a video clip.

Although Listing One is a short test program, it is complete. The implementation of the Start() and Stop() methods provides the basic functionality to control recording MPEG clips. Listing Two is a header file that contains a few simple utility routines, while the implementation (mpeg.h) is presented in Listing Three.

Database Design

I initially considered using an off-the-shelf database such as Excite (http://www.excite.com/), which is designed for indexing web pages. However, none of the available database engines were suitable for a real-time system. Without a real-time database, captured segments would not become immediately available. They would have to wait for the database to index them. The benefit of a custom realtime database is that access to stories is only limited by the time lag of system buffers in saving the data. Stories become available for search and retrieval almost immediately, even while still in the process of being captured.

Commercial databases are designed for generic uses, not for optimal speed on writes. Rather than use complicated &trees or object databases, the closed-captioned text database is based on simple flat filesit appends text and stores the length of each story. Conceptually, this is a simple design, but it is more difficult to edit data later when culling. Since the culling process has no real-time constraints, this limitation poses no significant problem. The alternative design, of myriad tiny files to contain each story, would have degraded keyword search performance.

Consequently, I built a custom search engine to scan through the closed-captioned text database looking for the specified word or phrase. As they are captured, the stories are numbered (1, 2, 3, and so on). The search function returns the number of the matching story so the story itself can be easily retrieved. The search engine was designed as a CGI scripta program that returns data through a web server.

Creating a C++ CGI Program There is considerable mystique about CGI and a common misconception that C++ is ill-suited to it. However, as Listing Four illustrates, it is straightforward to write a CGI program in C++.

For a CGI program, you simply output what you want the web browser to see using cout. Of course, it's a little difficult because your output is formatted as HTML code. You must also remember to output the proper "Content-Type" at the beginning, followed by two newlines. (Failing to do so may cause unpredictable errors from different web servers.) Under Windows, don't forget to compile your CGI as a console application.

As Listing Five shows, it's slightly more difficult (compared to Listing Four) to interact with an HTML form to get input from the user. Listing Six, the HTML for the form, instructs the web server that the program form_cgi should be executed in response to a user pressing Enter, and that the output of that program is what should be returned to the user's browser. You must, of course, have the web browser properly configured and the form_cgi executable in the web server's cgi-bin directory.

Although it may not seem like much, this (complete) example shows everything you need to know to work with a database search engine through a web interface. Instead of just displaying the query string back to the user as in this example, you would pass the query string through using a call into your database engine, then output the returned data to coat.

For a single-field input like this, you can use GET data. The field data is provided to you by the web server in the QUERY_STRING environment variable. More complicated forms require POST data, which is read off cin.

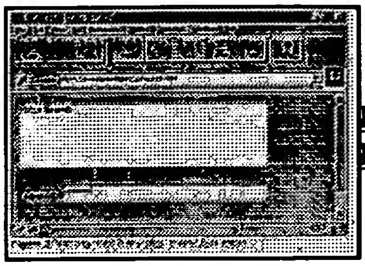
Porting to a Netscape Plug-In

Setting up a web server can be tricky, especially if you've never done it before. Each server has its own quirks, and there are hundreds of different servers to choose from on the various platforms. Don't forget that you must have TCP/IP networking set up properly first. That, of course, requires that your networking hardware is correctly installed. Shipping SVCR with only the CGI version of the search engine would be asking for a support nightmare. So, a Netscape plug-in that would stand alone was created- no web server required.

A plug-in lets you extend Netscape, usually for the purpose of writing an inline viewer of a new file format. The Netscape plug-in API supports Windows, Macintosh, and UNIX platforms. Under Windows, plug-ins are DLLs. The power of a plugin can be awesome. They can be called by a Java method. Once inside the C++ code of your plug-in, you are no longer under the normal restrictions of the Java applet security model. You are running in native code in Windows. Further, you can call back from C++ into Java, manipulating its GUI or data. What we're talking about is a Java/C++ mixed-language environment. You get the performance and power of C++ along with the portable GUI of Java- the best of both worlds.

Designing Netscape plug-ins feels like being in a lost world. There is a passing similarity to MFC, since plug-ins are a kind of framework. However, that's where the similarity to any normal C++ code ends. The Netscape plug-in developer's kit (available at http://www.netscape.com/) includes boilerplate code for a typical plug-in and the javah compiler. The javah compiler writes C/C++ wrappers for the methods in Java libraries. It isn't a compiler really, but a code generator. It works for both standard Java classes and for classes you create.

Although the implementation is completely different, using javah is conceptually similar to working with COM. It gives you a way to get a pointer to a Java object and manipulate it as though it was a C++ object.



Enlarge 200%
Enlarge 400%

Figure 3:

Figure 3 is a web page with both a plug-in and a Java applet. The plug-in is embedded in the area where it says "Video Server Search." The string is actually being displayed by the plug-in DLL using a Windows call. You can display whatever you want in the plug-in's window using the normal Windows API calls.

The "Search:" string and its data entry field are a Java applet, not an HTML form. When users enter a string into the field, the applet calls into the plug-in, which actually does the search and writes an HTML page (a file on the local file system). Finally, the applet loads that page into the browser using ShowDocument(). The HTML code in Listing Seven is virtually the same for any HTML code that uses a Java-aware plug-in. There are two differences from simple HTML that allow Java to talk to the plug-in: The plug-in is actually embedded into the page by EMBED. You can't just new the Java class that has the native methods implemented in a plug-in. It has to be embedded in the page. The second significant point is the magic MAYSCRIPT tag. The method for calling a plug-in object out of a page is to ask JavaScript for a handle to the plugin. Even though there isn't any JavaScript code in the page, the applet needs the MAYSCRIPT tag to be allowed to call the JavaScript interpreter and use JSObject.

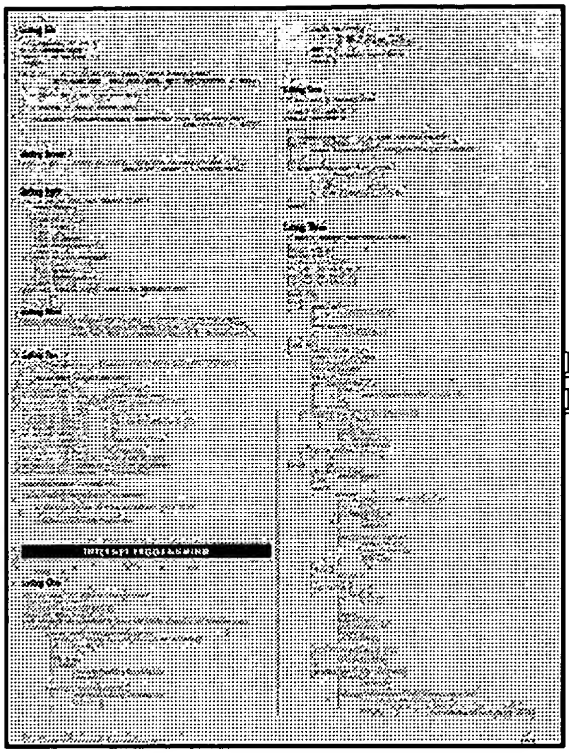
Instead of using new, you use code similar to Listing Eight, which fetches the JavaScript window, the document in the window, and the plug-in in the document. The name of the plug-in is the same as what it was called in the EMBED clause. Once the plug-in is retrieved it works like any other Java object. You call its native Search method, which hands off all the real work to your DLL. All of this code is just the normal plumbing necessary to use Java and C++ together in Netscape.

Conclusion

Anyone who wants to be able to produce a quick analysis of breaking news is a potential user of an SVCR. In addition to military and financial analysts, television broadcasters or film libraries wanting to search their own content or repurpose it for the Web are possible users, as are news clipping services.

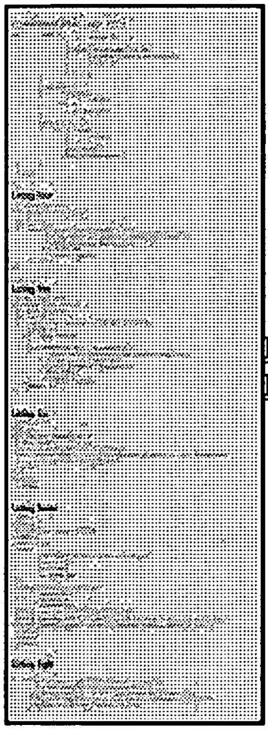
As the prices of multimedia hardware continue to tumble, I wouldn't be surprised if every new PC has the necessary hardware built in to support an SVCRwithin a few years. This could simplify development. We had a surprise recently when the Telescriber ceased production, necessitating migrating to a different closed-caption decoder.

We are starting to demonstrate prototypes at trade shows (such as AFCEA and DVExpo). Early adopters of our SVCR will include a top military command center and a major Wall Street brokerage house. Future enhancements will take us beyond U.S. television news monitoring to indexing other types of video information, both with and without closed captioning. To do that, we must add support for speech recognition and image understanding.



Enlarge 200%
Enlarge 400%

INTERNET PROGRAMMING



Enlarge 200%
Enlarge 400%

INTERNET PROGRAMMING

Enlarge 200%
Enlarge 400%

INTERNET PROGRAMMING

[Author note]

Robin is a program manager at a major systems integration company where he designs multimedia software. He can be contacted at rower@jumpsite.com.

Email Article ◁ Article 71 of 100 ▷ Publisher Info.

Print Article □ Mark article Article format: Cite/Abstract ▾

Save Link Saves this document as a Durable Link under "Results-Marked List"

Dawn of the Internet appliance

Computer [H.W. Wilson - AST]; Oct 1997; Lawton, George;

Volume:	30
Start Page:	16+
Page Count:	0
Document Type:	Feature Article
Source Type:	PERIODICAL
ISSN:	00189162
Subject Terms:	Web servers **Internet** protocols

Abstract:

*Developers can now incorporate **Internet** connectivity into a range of devices, including factory machinery, **VCRs**, and handheld appliances. The standardization of **Internet** protocols and competition among **Internet** service providers have lowered the price of **Internet** connectivity, prompting vendors to provide connectivity in a wide range of devices. As web browsers are so common, developers can provide sophisticated low-cost interfaces by supporting web servers in their products.*

ProQuest® Return to NPL Web Page ?Help

Collections | Search Methods ▾ | Topic Finder | Browse Lists ▾ | Results ▾ | Search Guide

Searching collection: All Collections Article Display

Email Article Article 75 of 100 Publisher Info.

Print Article □ Mark article **Article format:** Full Text ▾

Save Link Saves this document as a Durable Link under "Results-Marked List"

TWICE

Program Master TV/VCR system shipped by STI

Twice; New York; Aug 4, 1997; Greg Tarr;

Source (subtitle): this week in consumer electronics
Volume: 12
Issue: 18
Start Page: 28
ISSN: 08927278
Subject Terms: Television
Video **recorders**
Consumer electronics
Internet
Software
Companies: Scottsdale Technologies

Abstract:
*Scottsdale Technologies is now shipping its Program Master TV and **VCR** record programming system that uses an **Internet**-delivered TV schedule service to automatically tune TVs, **VCRs** and cable boxes through a unique IR remote control device.*

Full Text:

After a year-long software development delay, Scottsdale, Ariz.-based Scottsdale Technologies Inc. (STI) said it is now shipping its Program Master TV and VCR record programming system that uses an Internet-delivered TV schedule service to automatically tune TVs, VCRs and cable boxes through a unique IR remote control device.

The company said it is using computer hardware distributor Tech Data to market the product to CE and computer retail accounts. Best Buy and CompUSA "will be the first to receive" the product, and the company "anticipates that Program Master will be available in all major retail stores nationwide by the end of the summer," said Michael Slover, STI marketing director.

"Unfortunately, software development can be tricky, and it took us a little longer to deliver the product than we expected, but the good news is that the delay gave us time to add some nice additional features," Slover said.

Among new feature enhancements is the ability to control most brands of satellite receiver boxes, he said.

Program Master ($49.95 expected street price) is an oval palm-size IR remote control equipped with a single button. The remote handset is designed to record data delivered through the TV Host Web site via light signals reflected through a computer monitor.

TV Host is a subscription-based service providing both detailed TV program schedules tailored to user preferences and IR program codes for a virtually limitless number of consumer electronics products.

The service offers program titles with start and stop times, along with plot descriptions, ratings and actors. Users perform key word searches to call up programs affiliated with a wide range of subjects and categories.

After identifying a list of programs a user wants to view or record, he or she holds the remote up to the PC monitor and clicks on the program selection, and data is relayed by light into the remote's memory.

The device "remotely instructs the user's TV, VCR, cable box and satellite receiver when to power on, what channel to select, and when to start and end recording television programs the user has preselected for viewing and/or recording," STI said.

The Program Master remote is designed to be upgraded through the Internet with IR codes for current and future remote-controllable devices.

"This is the first product that actually marries a home entertainment system to the PC, even if the PC is in another room in the house," said Slover.

TV Host is an independently owned company. STI and retailers share in the subscription revenue stream.

Purchasers are given a free trial subscription to TV Host for six months, after which subscription fees range from $30 for six months to 550 for a year. - Greg Tarr

ProQuest® Return to NPL Web Page ?Help

Collections | Search Methods ▾ | Topic Finder | Browse Lists ▾ | Results ▾ | Search Guide

Searching collection: All Collections Article Display

Email Article | Article 29 of 50 | Publisher Info.

Print Article | □ Mark article | Article format: Text+Graphics ▾

Save Link Saves this document as a Durable Link under "Results-Marked List"

Invasion of the embedded systems

Network World; Framingham; Jun 8, 1998; Kimberly Patch; Eric Smalley;

Duns: **00-912-2532**	
Volume:	15
Issue:	23
Start Page:	1,53
ISSN:	08877661
Subject Terms:	Embedded systems Computer networks **Remote control** Electronics Energy management
Classification Codes:	**9190**: *US* **5230**: *Hardware* **5250**: *Telecommunications systems* **5150**: *Energy management*
Geographic Names:	US
Companies:	University of North Dakota Hewlett Packard Co*Ticker:* **HWP** *Duns:* **00-912-2532**

Abstract:

Nearly everything that uses electricity now incorporates semiconductors. These devices are increasingly being connected to corporate networks and the ***Internet****. One of the main benefits motivating companies to integrate devices into their networks is cost recovery. Every device that consumes or dispenses something requires human beings to replenish its supply. Attaching those devices to the network makes it possible to see in real time exactly what device needs resupplying and when. Software technologies are emerging that help enable this communication with various devices. Java Card lets developers write applets that run on smart cards. Embedded Java applications run on real-time operating systems on 8- to 32-bit embedded processors. Personal Java is aimed at devices such as handheld PCs and set-top boxes. Hewlett-Packard Co.'s JetSend is a communications protocol intended to work over any transport and on any device.*

Full Text:

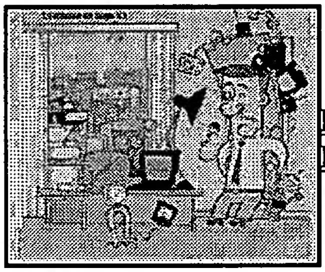
You may think you'll be able to catch your breath In a couple of years. After all, you're beginning to get a handle on remote access and electronic commerce, and the Year 2000 problem has a builtin deadline. But lurking on the horizon Is a new challenge that will keep you hopping long after the new millennium arrives, potenUally adding whole new classes of devices to your network.

The first phase of this Insidious plot is almost complete; nearly everything that uses electricity, from light fixtures to copy machines to air conditioners, now Incorporates semiconductors. And Phase 2 is under way as increasingly these devices are being connected to corporate networks and the Internet. The Invasion of the embedded systems Is at hand.

The number of devices that could be attached to a network is impressive: stand-alone office equipment such as photocopiers; input devices such as cameras and microphones; road warrior gear including handheld PCs, cell phones and pagers; transaction-oriented devices, from vending machines to card readers; environmental sensors such as thermostats and pressure gauges; building systems such as lighting monitors and door locks; and consumer electronics devices including toys, set-top boxes and game consoles.

The bottom line is networks are extending their reach ever farther. This poses some obvious problems, including heavier traffic, burgeoning management data and new types of security threats. But the coming world of net-attached devices also harbors opportunities.

Networks are strategic because they facilitate information sharing. The coming boom of netenabled devices will expand that mission to include real-time information about inventory, resource consumption, maintenance status and human traffic patterns. This information will translate directly into more efficient use of resources and personnel, and facilitate just-intime processes. The question is, will you be overwhelmed by the burden of integrating all these devices before you can tap this wealth of data?



Enlarge 200%
Enlarge 400%

One of the main benefits motivating companies to integrate devices into their networks is cost recovery, according to John Gage, chief scientist at Sun Microsystems, Inc.'s Sun Labs in Mountain View, Calif. Every device that consumes or dispenses something, whether it's toner or a soft drink, requires human beings to replenish its supply. Attaching those devices to the network makes it possible to see in real time exactly what device needs resupplying and when. This allows companies to manage costly personnel time more effectively.

"I think the future of networking is the future of efficiencies," Gage says.

The physical plant department at the University of North Dakota in Grand Forks has devices that measure temperature, pressure, wind speed, wind direction and carbon monoxide levels all around the campus. The data is transported over the campuswide NetWare network to building management and environmental control applications running on a Windows NT server (see graphic, below).

In this fashion, the university's boiler management system, in effect, can communicate with its air-conditioning system, says Randy Bohlman, technology advancement coordinator for the department. Gone are the days when the boiler could kick in on a cool September morning causing the air conditioner to work harder. Likewise, the university now has no problem keeping a specific room cool, such as a data center, while surrounding rooms are comfortably warm.

By integrating its boiler management system with its environmental control system, the university has been able to save 25% of its energy costs, or about $750,000 annually, Bohlman says.

"We crunch numbers [coming from environmental sensors] in different algorithms for energy savings, and then the results of this work go out to the individual controllers," Bohlman says. "We can economically make these interconnects [using the network] where they [would otherwise] cost hundreds of thousands of dollars."

Users are able to access environmental data from any workstation on campus and can easily pull it into spreadsheets to generate reports. This accounts for significant labor savings, Bohlman adds.

The university achieves additional labor savings because "we're using a lot of neural intelligence within our building management and control network," Bohlman says. The environmental control system "tunes itself. It automatically begins to compensate and correct for alterations in the various systems," he says. For example, the system recalibrates itself when a new sensor comes online.

There is potential for cost savings in network systems as mundane as lighting. The British Columbia Institute of Technology in Vancouver saves 20% to 30% on lighting costs through a combination of sensors that feed a centralized control system, says Gil Moore, physical plant support services manager. Sensors determine how much natural light is in a room, and the control system adjusts the level of electric light accordingly. Similarly, there are sensors that can determine when nobody is in a room, prompting the control system to shut off the lights. The institute also has granular control over the level of light in different rooms so, for example, it can ensure there's always ample light in a conference room. And all this is supported over the school's campus data network.

Teaching devices to talk

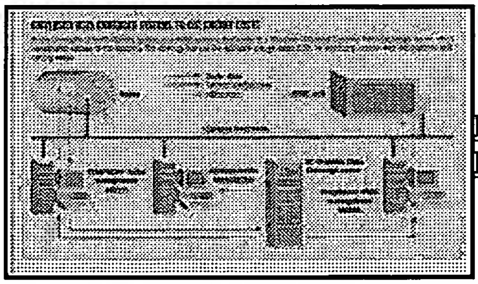
Beyond the monitoring and rudimentary control capabilities of SNMP, software technologies are emerging that help enable this communication with various devices. The technologies include Sun's Java variants, Hewlett-Packard Co.'s JetSend and Lucent Technologies, Inc.'s Inferno.

From the Java camp, there's Java Card, which lets developers write applets that run on smart cards. Embedded lava applications run on realtime operating systems on 8- to 32-bit embedded processors. Personal Java is aimed at devices such as handheld PCs and set-top boxes.

HP's JetSend is a communications protocol intended to work over any transport and on any device. It classifies devices ranging from PCs to digital cameras as "senders" and "receivers" and allows them to communicate without knowing the properties of the other device, according to Jim Hammons, HP's Alliance Programs manager.

The software allows two devices - for instance, a digital camera in New York and a photocopier in Los Angeles - to talk to each other over a network, with no computer or additional software required. Similar to the way modems negotiate data rates,JetSend negotiates the best common information format among devices such as fax machines, personal digital assistants (PDA), photocopiers, digital cameras, scanners and smart cards. JetSend is designed for embedded systems, operating in as little as 200K for IP devices and 50K for devices using simpler transports such as infrared, Hammons says.

New York brokerage house Bear, Stearns & Company, Inc. is testing JetSend for use in its IPand Windows NT-based network, according to an associate director for the company, who asked not to be identified. Bear, Stearns is looking for JetSend to ease the integration of networkattached devices such as scanners and plans to extend use of the technology to PDAs and eventually even photocopiers, he says.



Enlarge 200%
Enlarge 400%

UNIVERSITY USES EMBEDDED SYSTEMS TO CUT ENERGY COSTS